SECRET KEY GENERATION METHOD, ENCRYPTION METHOD, CRYPTOGRAPHIC COMMUNICATIONS METHOD, COMMON KEY GENERATOR, CRYPTOGRAPHIC COMMUNICATIONS SYSTEM, AND RECORDING MEDIA

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to a secret key generation method for generating secret keys peculiar to entities, to an encryption method for encrypting information so that it will be unintelligible to any but an authorized party, and to a cryptographic communications method which performs communications with ciphertext.

Description of the Related Art

In today's world, characterized by sophisticated information utilization, important business documents and image information are transmitted and processed in the form of electronic information over an infrastructure of computer networks. By its very nature, electronic information can be easily copied, making it extremely difficult to distinguish between the copy and the

original, and information security has become a very serious problem. The realization of computer networks which support "shared computer resources," "multi-access," and "broad-area implementation" is particularly indispensable to the establishment of a high-level information society. However, that very realization involves aspects which are inconsistent with the security of information exchanged between authorized parties. An effective technique for eliminating that inconsistency is encryption technology, which up until now, in the course of human history, has been primarily used in the fields of military operations and foreign diplomacy.

Cryptography is the process of exchanging information so that its meaning cannot be understood by anyone other than the authorized parties. In cryptographic operations, the conversion of the original text (plaintext) that anyone can understand to text (ciphertext) the meaning of which cannot be understood by a third party is called encryption, and the restoration of the ciphertext to plaintext is called decryption. The overall system wherein this encryption and decryption are performed is called a cryptosystem. In the processes of encryption and decryption, respectively, secret information called encryption keys and decryption keys are employed. A secret decryption key is necessary at the time of decryption, wherefore only a party

knowledgeable of that decryption key can decrypt the ciphertext. Accordingly, the confidentiality of the information is maintained by the encryption.

The encryption key and decryption key may be the same or they may be different. A cryptosystem wherein both keys are the same is called a common key cryptosystem, and the DES (Data Encryption Standards) adopted by the Bureau of Standards of the U.S. Department of Commerce is a typical example thereof. Conventional examples of such common key encryption schemes can be divided into the following three types.

(1) Type 1

Methods wherewith all common keys to be shared with possible parties in cryptographic communications are held in secret.

(2) Type 2

Methods wherewith keys are mutually shared by a preparatory communication each time cryptographic communications are conducted (including Diffie-Hellman-based key sharing scheme, key distribution scheme based on public key schemes, etc.).

(3) Type 3

Methods wherewith disclosed identification information (ID information)) that specifies an individual, such as user (entity) name and address, etc., is used, and both the sending entity and receiving entity

independently generate the same common key without preparatory communications (including KPS (key predistribution systems), ID-NIKS (ID-based non-interactive key sharing schemes), etc.).

Such conventional methods as seen in these three types of schemes are subject to the problems described below. With method 1, since all of the common keys are stored, this scheme is unsuitable for a network society wherein an unspecified large number of users become entities and conduct cryptographic communications. With method 2, there is a problem in that preparatory communications are required for key sharing.

Method 3 is a convenient method because it requires no preparatory communications, and a common key with any opposite party can be generated using the disclosed ID information of the opposite party together with characteristic secret parameters distributed beforehand from a center. Nevertheless, this scheme is subject to the following two problems. Firstly, the center must become a "big brother" (creating a key escrow system wherein the center holds the secrets of all of the entities). Secondly, there is a possibility that some number of entities could collude to compute the center secrets. In the face of this collusion problem, many innovative techniques have been devised to circumvent the

problem by way of computation volume, but a complete solution is very difficult.

The difficulties of resolving this collusion problem arise from the fact that the secret parameters based on the ID information form dual structures comprising center secrets and personal secrets. With method 3, a cryptosystem is configured using the disclosed parameters of the center, the disclosed ID information of the individual entities, and the two types of secret parameters for the center and entities. Not only so, but it is necessary also to configure such that center secrets will not be revealed even if the entities compare the personal secrets distributed to each. Accordingly, there are many problems that must be resolved before this cryptosystem can be actually realized.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a secret key generation method, encryption method, and cryptographic communications method based on an ID-NIKS, wherewith specifying information (ID information) is divided into a plurality of portions, and all secret keys based on the divided specifying information are distributed to entities from each of a plurality of centers, whereby it is possible to minimize the

mathematical structures, circumvent the collusion problem, and facilitate the construction of the cryptosystem.

Another object of the present invention is to provide a secret key generation method, encryption method, and cryptographic communications method that are more highly resistant to random number substitution attack.

According to a first aspect of the present invention, there is provided a secret key generation method for generating secret keys peculiar to entities that are to be sent from a center to the entities, characterized in that the secret keys peculiar to the entities are generated using divided specifying information resulting from the division of information specifying the entities.

According to a second aspect of the present invention, there is provided an encryption method wherein secret keys peculiar to entities are sent to the entities from the center respectively, and an entity encrypts plaintext to ciphertext using a secret key peculiar to that entity sent from the center, characterized in that the secret keys peculiar to the entities are generated using divided specifying information resulting from the division of information specifying the entities, and plaintext is encrypted to ciphertext at one entity that is a ciphertext sender using a common key generated from a component contained in its own secret key, the component corresponding to the divided specifying

information of another entity that is a destination of the ciphertext.

According to a third aspect of the present invention, there is provided a cryptographic communications method for communicating information between entities, wherein one entity encrypts plaintext to ciphertext using a first common key derived from a first secret key peculiar to that entity sent from a center and sends the ciphertext to another entity (recipient), and the recipient decrypts the ciphertext to the plaintext using a second common key identical to the first common key, the second common key being derived from a second secret key peculiar to the recipient sent from the center, characterized in that a plurality of the centers are deployed, each of the centers generates secret keys peculiar to the entities using divided specifying information resulting from the division of information specifying the entities, and each of the entities generates the common key using a component, contained in its own secret key, corresponding to the divided specifying information of an opposite entity.

The reason why the various cryptosystems based on entity specifying information proposed for the purpose of resolving the collusion problem have been unsuccessful lies in excessively seeking mathematical structures to provide innovative techniques for preventing center

secrets from being deduced from entity collusion information. When the mathematical structures are too complex, the method of demonstrating safety becomes very difficult. In the present invention, therefore, the mathematical structures are held to a bare minimum by dividing entity specifying information into a plurality of portions and distributing all the secret keys for each of the divided specifying information to the entities.

In the present invention, a plurality of centers are deployed, and each center generates a secret key corresponding to one unit (or piece) of divided specifying information for one entity. Accordingly, no single center holds all of the entity secrets and hence no center becomes a "big brother." Also, because the mathematical structures are held down to a minimum, circumvention of the collusion problem is easily realized and the cryptosystem is also simple to implement. Furthermore, the secret keys peculiar to one entity for that entity to generate a common key have been sent from the centers and are stored from the start in table form, wherefore the time required for common key generation can be significantly shortened.

According to a fourth aspect of the present invention, there is provided a secret key generation method for generating secret keys specific to entities using divided specifying information resulting from the

division of information specifying the entities into a plurality of blocks, characterized in that the secret key for a first block of divided specifying information has a multi-layer structure and each of the secret keys for the remaining blocks of divided specifying information has a single-layer structure.

According to a fifth aspect of the present invention, there is provided an encryption method wherein secret keys peculiar to entities are generated using divided specifying information resulting from the division of information specifying the entities into a plurality of blocks, plaintext is encrypted to ciphertext using a common key generated using a component, contained in the secret key, corresponding to the divided specifying information for an opposite entity to which the ciphertext is to be sent, characterized in that the secret key for a first block of divided specifying information has a multi-layer structure, and each of the secret keys for the remaining blocks of divided specifying information has a single-layer structure.

According to a sixth aspect of the present invention, there is provided a cryptographic communications method for communicating information between entities, wherein a plurality of centers are deployed, each of which generates secret keys peculiar to the entities using divided specifying information resulting from the

division of information specifying the entities into a plurality of blocks, one entity generates a first common key using a first component contained in secret keys peculiar to that entity sent from the centers and corresponding to the divided specifying information of another entity (recipient), encrypts plaintext to ciphertext using the first common key, and sends the ciphertext to the recipient, the recipient generates a second common key identical to the first common key, using a second component contained in secret keys peculiar to the recipient sent from the centers and corresponding to the divided specifying information of the ciphertext sender, and decrypts the ciphertext to the original plaintext using the second common key, the secret key for a first block of divided specifying information has a multi-layer structure, and the secret keys for the remaining blocks of divided specifying information have a single-layer structure.

The present invention is configured in such a manner that the common key can only be derived after the computation for all blocks is complete, and a divided block of information specifying a specific entity cannot be attacked independently, whereupon random number substitution attack can be circumvented.

The term "recording medium" or "computer usable (or readable) medium" in this specification includes any

physical object in which a program to be executed by CPU or the like is stored. For example, the "recording medium" includes a floppy disc, CD-ROM, hard disk drive, ROM, RAM, optical recording medium such as DVD, photo-magnetic recording medium such as MO, magnetic recording medium such as magnetic tape, and semiconductor memory such as IC card and miniature card. A data signal embodied in a carrier wave may be the computer readable medium.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates a model diagram representing the configuration of a cryptographic communications system of the present invention;

Fig. 2 illustrates a model diagram representing an example of entity ID vector division;

Fig. 3 illustrates a model diagram showing how information is communicated between two entities;

Fig. 4 is a diagram representing the configuration of another cryptographic communications system according to the present invention;

Fig. 5 depicts another example of entity ID vector division;

Fig. 6 is a diagram showing how information is communicated between two entities; and

Fig. 7 is a diagram showing the configuration of a recording media.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments of the present invention are now described.

Fig. 1 is a model diagram representing the configuration of an cryptographic communications system of the present invention. A plurality of centers 1 (K in number) which can be trusted to maintain information confidentiality are established. These centers 1 may be public institutions in a society, for example. The deployment of the plurality of centers 1 is the point of difference with the conventional third method.

These centers 1 are connected to a plurality of entities a, b, ..., z that are the users employing this cryptosystem by secret channels (communication paths) 2_{a1}, 2_{aK}, 2_{b1}, ..., 2_{bK}, ..., 2_{z1}, ..., 2_{zK}. Secret information is sent from the centers 1 via these secret communication paths to the entities a, b, ..., z. Communication paths 3ab, 3az, 3bz, etc., are also provided between pairs of entities. Ciphertext obtained by encrypting

communications information is sent back and forth between entities via these communication paths 3ab, 3az, 3bz, etc.

1st Embodiment:

A first embodiment that is a basic scheme of the present invention is described first.

Preparatory processing at centers 1:

The centers 1 prepare public keys and secret keys as follows and disclose the public keys.

Public key	P	Large prime number
	L	Size of ID vector (L = KM)
	K	Number of ID vector division blocks
	M	Size of divided ID vector
Secret key	g	GF (P) primitive element
	H_j	Symmetrical $2^M \times 2^M$ matrix formed of random numbers ($j = 1, 2, \ldots, K$)
	α_{ij}	Personal secret random number of entity i (where $\alpha_{i1}\alpha_{i2} \ldots \alpha_{iK} \equiv 1 \pmod{P-1}$)

ID vectors that are specifying information indicating the names and addresses of entities are made L-dimension binary vectors, and each of the ID vectors is divided into K blocks (each has a block size M) as diagrammed in Fig. 2. The ID vector for entity i (i.e.

vector I_i), for example, is divided as indicated in formula 1 below. The vectors I_{ij} (j = 1, 2, ..., K), that are divided specifying information, are called ID division vectors.

(1) $$\overrightarrow{I_i} = [\overrightarrow{I_{i1}}|\overrightarrow{I_{i2}}|\cdots|\overrightarrow{I_{iK}}]$$

Entity registration processing:

When each of the centers 1 is requested by an entity i for registration, K secret vectors s_{ij} (j = 1, 2, ..., K) corresponding, respectively, to a prepared key and K ID division vectors for entity i are found according to formulas 2-1, 2-2, ..., 2-K, as represented below, the vectors s_{ij} so found are sent to entity i in secret, and registration is complete.

(2-1) $$\overrightarrow{s_{i1}} \equiv g^{\alpha_{i1} H_1[\overrightarrow{I_{i1}}]} \pmod{P}$$

(2-2) $$\overrightarrow{s_{i2}} \equiv \alpha_{i2} H_2[\overrightarrow{I_{i2}}] \pmod{P-1}$$

$$\vdots$$

(2-K) $$\overrightarrow{s_{iK}} \equiv \alpha_{iK} H_K[\overrightarrow{I_{iK}}] \pmod{P-1}$$

However, when g is a scalar, and A and B are matrixes, the representation $B = g^A$ indicates that power multiplication on g is performed for each component (μ, ν) of A. In other words, the result is as given in formula 3 below. The representation H_j [vector I_{ij}]

indicates that one row corresponding to the vector I_{ij} is extracted from the symmetrical matrix H_j, and the $[\cdot]$ operation is also defined for reference.

(3) $$B_{\mu\nu} = g^{A_{\mu\nu}}$$

Processing for generating common keys between entities:

Entity i selects from its own secret key vectors s_{i1} a vector s_{i1} [vector I_{m1}] of the component corresponding to vector I_{m1} that is the ID division vector of entity m, and also selects from among the secret key vectors s_{ij} for each of the blocks j (j = 2, ..., K) the vector s_{ij} [vector I_{mj}] of the component corresponding to the vector I_{mj}. Then, entity i sequentially power-multiplies all of the vectors s_{ij} [vector I_{mj}] (j = 2, ..., K) except for the vector s_{i1} [vector I_{m1}], with modulo P and the vector s_{i1} [vector I_{m1}] as the base, thereby deriving the common key K_{im}. The computation formula for finding this K_{im} specifically becomes formula 4 below. This K_{im} coincides with the common key K_{mi} derived at the entity m end.

(4)
$$\begin{aligned} K_{im} &\equiv \overrightarrow{s_{i1}}[\overrightarrow{I_{m1}}]^{\overrightarrow{s_{i2}}[\overrightarrow{I_{m2}}]\cdots\overrightarrow{s_{iK}}[\overrightarrow{I_{mK}}]} \\ &\equiv g^{\alpha_{i1}\cdots\alpha_{iK}\cdot H_1[\overrightarrow{I_{i1}}][\overrightarrow{I_{m1}}]\cdots H_K[\overrightarrow{I_{iK}}][\overrightarrow{I_{mK}}]} \\ &\equiv g^{H_1[\overrightarrow{I_{i1}}][\overrightarrow{I_{m1}}]\cdots H_K[\overrightarrow{I_{iK}}][\overrightarrow{I_{mK}}]} \pmod{P} \end{aligned}$$

Next, the communication of information between entities in the cryptosystem described above is described. Fig. 3 illustrates information communicated between two entities a and b. In the example diagrammed in Fig. 3, entity a encrypts a plaintext (message) M to a ciphertext C which it sends to entity b, and entity b decrypts that ciphertext C back to the original plaintext (message) M.

A secret key generator 1a is provided at the j'th center 1 (where j = 1, 2, ..., K) for deriving the vectors s_{aj} and s_{bj} (secret keys) peculiar to the entities a and b, respectively, following formula 2-j given earlier. Then, when a request for registration is tendered from the entities a and b, the secret key vectors s_{aj} and s_{bj} for those entities a and b are sent to the entities a and b.

Entity a is provided with a memory 10 for storing, in tabular form, the characteristic secret key vectors s_{a1}, ..., s_{aj}, ..., s_{aK} sent from the K centers 1, a component selector 11 for selecting from among those secret key vectors the vector s_{a1} [vector I_{b1}], ..., vector s_{aj} [vector I_{bj}] ..., vector s_{aK} [vector I_{bK}] for the components corresponding to entity b, a common key generator 12 for generating the common key K_{ab} with entity b sought by entity a using those components selected, and an encryptor 13 for encrypting the plaintext (message) M

to the ciphertext C using that common key K_{ab} and outputting it over the communication path 30.

Entity b, meanwhile, is provided with a memory 20 for storing, in tabular form, the characteristic secret key vectors s_{b1}, ..., s_{bj}, ..., s_{bK} sent from the centers 1, a component selector 21 for selecting from among those secret key vectors the vector s_{b1} [vector I_{a1}], ..., vector s_{bj} [vector I_{aj}], ..., vector s_{bK} [vector I_{aK}] for the components corresponding to entity a, a common key generator 22 for generating the common key K_{ba} with entity a sought for by entity b using those components selected, and a decryptor 23 for decrypting the ciphertext C input from the communication path 30 to the plaintext (message) M using that common key K_{ba} and outputting it.

When information is to be sent from entity a to entity b, first, the secret key vectors s_{a1}, s_{a2}, ..., s_{aK} pre-stored in the memory 10 after being derived according to the formulas 2-1, 2-2, ..., 2-K at the centers 1 are read out to the component selector 11. Then, the component selector 11 selects the vector s_{a1} [vector I_{b1}], vector s_{a2} [vector I_{b2}], ..., and vector s_{aK} [vector I_{bK}] that are the components corresponding to entity b, and sends them to the common key generator 12. The common key generator 12 uses these components to derive the common key K_{ab} according to formula 4, and sends that common key K_{ab} to the encryptor 13. With the encryptor 13,

this common key K_{ab} is used to encrypt the plaintext (message) M to the ciphertext C and the ciphertext C is sent via the communication path 30.

The ciphertext C sent over the communication path 30 is input to the decryptor 23 of entity b. The secret key vectors s_{b1}, s_{b2}, ..., s_{bK} derived according to formulas 2-1, 2-2, ..., 2-K at the centers 1 and prestored in the memory 20 are read out to the component selector 21. Then, the component selector 21 selects the vector s_{b1} [vector I_{a1}], vector s_{b2} [vector I_{a2}], ..., vector s_{bK} [vector I_{aK}] that are components corresponding to entity a, and sends them to the common key generator 22. The common key generator 22 uses these components to derive the common key K_{ba} according to formula 4 and sends this common key K_{ba} to the decryptor 23. The decryptor 23 uses the common key K_{ba} to decrypt the ciphertext C to the plaintext (message) M.

In the scheme of the present invention, the secret key vectors peculiar to the entities are stored beforehand in the memories of the entities so that a shorter time is required to generate the common keys.

The safety provided by the scheme of the present invention is now discussed.

It is known that one of the conditions necessary to a safe ID-NIKS is the inability of separating the secret key generating functions and key sharing functions in

polynomial time. A fact that the scheme of the present invention satisfies this necessary condition is described below.

Secret key generating function:

The scheme of the present invention has a total of K secret key generating functions as indicated in formulas 5 and 6 below.

$$f_1(\vec{x}) = g^{\alpha_{i1}} H_1[\vec{x}] \quad (j = 1) \tag{5}$$

$$f_j(\vec{x}) = \alpha_{ij} H_j[\vec{x}] \quad (j = 2, \ldots, K) \tag{6}$$

If H is an arbitrary symmetrical matrix, then the referencing function $[\cdot]$ is clearly indivisible, as shown in formulas 7 and 8 below.

$$H[\vec{x} + \vec{y}] \neq H[\vec{x}] + H[\vec{y}] \tag{7}$$

$$H[\vec{x} + \vec{y}] \neq H[\vec{x}] \cdot H[\vec{y}] \tag{8}$$

Thus, the K secret key generating functions represented in formulas 5 and 6 are indivisible, as shown in formula 9 below.

$$f_j(\vec{x} + \vec{y}) \neq f_j(\vec{x}) \circ f_j(\vec{y}) \quad (j = 1, 2, \ldots, K) \tag{9}$$

Key sharing function:

The key sharing function in the scheme of the present invention is represented in formula 10 below.

(10) $$\mathcal{F}(\vec{x}, \vec{y}) = g^{H_1[\vec{x_1}][\vec{y_1}]\cdots H_K[\vec{x_K}][\vec{y_K}]}$$

As in the case of the secret key generating functions, the key sharing function represented in formula 10 is indivisible, as shown in formula 11 below.

(11) $$\mathcal{F}(\vec{a}, \vec{x} + \vec{y}) \neq \mathcal{F}(\vec{a}, \vec{x}) \circ \mathcal{F}(\vec{a}, \vec{y})$$

Attacks for breaking cryptosystems by the collusion of an indefinite number of entities (hereinafter "non-corrupting collusion") has been debated for quite some time. At the same time, attacks conducted by a smaller number of collaborators wherein only entities necessary for the attack are bought (hereinafter "corrupting collusion") are also effective if a certain individual is the only target. The safety of the scheme of the present invention against such corrupting and non-corrupting collusions is now considered.

Safety against non-corrupting collusion:

In cases where it is possible to represent the ID vector of any entity by a linear combination of collaborator ID vectors (combination attack) and either the secret key generating function or key sharing function is divisible in polynomial time, it is possible to counterfeit the secret keys of other entities from the secret keys of the collaborators (separation attack). Such an attack is known as a linear attack.

In the scheme of the present invention, the ID vector of any entity can be represented as a linear combination by using the ID vectors of L collaborators who are linearly independent. That is, a combination attack by L or more entities is viable. However, because the secret key generating functions and key sharing function are indivisible functions, as noted earlier, the secret key and common key of that entity cannot be counterfeited by a separation attack even in the unlikely case where a combination attack against any entity should become viable. Therefore the linear attack does not work with the scheme of the present invention. Accordingly, in the face of a non-corrupting collusion, the scheme of the present invention has a collusion threshold (minimum number of collaborators required for combination attack) that is far higher than L.

Safety against corrupting collusion:

In cases where an attack is made against the scheme of the present invention wherein a specific entity is targeted, a random number substitution attack like that described below is conceivable wherein all of the entities required for the attack are bought out and all of the secret keys of the bought-out entities are used.

The situation is described in an example where the name is four Kanji characters (L = 4 × 16 = 64 bits) so that the entity ID is easy to understand and each Kanji

character is treated as 1 block. In other words, it is assumed that K = 4 and M = 16.

A case is now considered wherein the IDs of entities Z, A, B, C, and D are set as noted below, entities A, B, C, and D are bought out, and entity Z is attacked.

$$\begin{aligned}
\overrightarrow{I_Z} &= [辻 | 井 | 重 | 男] \\
\overrightarrow{I_A} &= [辻 | 本 | 恵 | 子] \\
\overrightarrow{I_B} &= [中 | 井 | 邦 | 夫] \\
\overrightarrow{I_C} &= [山 | 田 | 重 | 人] \\
\overrightarrow{I_D} &= [佐 | 藤 | 和 | 男]
\end{aligned}$$

The secret key of entity Z is then given as follows.

$$\begin{aligned}
\overrightarrow{s_{Z_1}} &\equiv g^{\alpha_{Z1} H_1[辻]} \pmod{P} \\
\overrightarrow{s_{Z_2}} &\equiv \alpha_{Z2} H_2[井] \pmod{P-1} \\
\overrightarrow{s_{Z_3}} &\equiv \alpha_{Z3} H_3[重] \pmod{P-1} \\
\overrightarrow{s_{Z_4}} &\equiv \alpha_{Z4} H_4[男] \pmod{P-1}
\end{aligned}$$

The collaborators make the following computations and counterfeit the secret key of entity Z.

$$\begin{aligned}
\overrightarrow{s_{Z_1}}' &\equiv \overrightarrow{s_{A_1}} \equiv g^{\alpha_{A1} H_1[辻]} \pmod{P} \\
\overrightarrow{s_{Z_2}}' &\equiv \frac{\overrightarrow{s_{A_2}}[井]}{\overrightarrow{s_{B_2}}[本]} \cdot \overrightarrow{s_{B_2}} \equiv \frac{\alpha_{A2} H_2[本][井]}{\alpha_{B2} H_2[井][本]} \cdot \alpha_{B2} H_2[井] \\
&\equiv \alpha_{A2} H_2[井] \pmod{P-1} \\
\overrightarrow{s_{Z_3}}' &\equiv \frac{\overrightarrow{s_{A_3}}[重]}{\overrightarrow{s_{C_3}}[恵]} \cdot \overrightarrow{s_{C_3}} \equiv \frac{\alpha_{A3} H_3[恵][重]}{\alpha_{C3} H_3[重][恵]} \cdot \alpha_{C3} H_3[重] \\
&\equiv \alpha_{A3} H_3[重] \pmod{P-1}
\end{aligned}$$

$$\overrightarrow{s_{Z_4}}' \equiv \frac{\overrightarrow{s_{A_4}}[男]}{\overrightarrow{s_{D_4}}[子]} \cdot \overrightarrow{s_{D_4}} \equiv \frac{\alpha_{A4}H_4[子][男]}{\alpha_{D4}H_4[男][子]} \cdot \alpha_{D4}H_4[男]$$
$$\equiv \alpha_{A4}H_4[男] \pmod{P-1}$$

It may be seen here that the counterfeited vectors s_{z1}' to s_{z4}' work in the same manner as the vectors s_{z1} to s_{z4}, respectively. Hence the collusion attack is definitely viable against the scheme of the present invention in situations where it is possible to buy out enough entities to mount the attack.

In order for this corrupting collusion attack to be viable, however, it is necessary to acquire the secret keys of a collaborator having exactly the same ID division vectors as the K number of ID division vectors of the entity targeted for attack. For some specific block, only one entity in 2^M entities has exactly the same ID division vectors. Buying all of the K blocks for this special entity, even assuming the values M = 10 and K = 100, is hardly an easy task. Accordingly, the scheme of the present invention may be said to be safe against corrupting collusions. The parameters M and K can be suitably set according to the scale of the cryptosystem and/or to the degree of safety required.

Now, in order to circumvent a random number substitution attack by corrupting collusion, it is only necessary to implement measures to prevent the division

blocks from being independently attacked. In other words, it is only necessary to make it so that the random number terms disappear only after the computation of all of the blocks is complete. With this perspective, two embodiment are now described which represent improvements of the first embodiment.

2nd Embodiment:

Another example of the present invention (2nd embodiment) is now described which is made stronger against random number substitution attack by combining a random number elimination method.

Preliminary processing at centers 1:

As in the first embodiment, the centers 1 prepare public keys and secret keys as follows and disclose the public keys.

Public key	P	Large prime number
	L	Size of ID vector (L = KM)
	K	Number of ID vector division blocks
	M	Size of divided ID vector
Secret key	g	GF (P) primitive element
	H_j	Symmetrical $2^M \times 2^M$ matrix formed of random numbers ($j = 1, 2, \ldots, K$)
	α_i	Personal secret random number of

entity i

(where $\alpha_{i1}\alpha_{i2} \ldots \alpha_{iK} \equiv 1$

(mod P - 1))

In order to employ the safety of RSA ciphers, P is set so that it is very difficult to factor P - 1 into prime numbers. To do that it is only necessary to use a prime number such that P = 2pq + 1 (where p and q are prime).

As in the first embodiment, the ID vector of each of the entities is divided into K blocks (ID division vectors) having a block size M (cf. Fig. 2 and formula 1).

Furthermore, as indicated in formula 12 below, a hashing function h(·) for generating a second ID vector v_i of K-1 dimension from the ID is disclosed by the centers 1. The components of this second ID vector v_i generated with the hashing function take positive integers, and it is assumed that the sum thereof is a comparatively small constant e as represented in formula 13 below.

(12)

$$\overrightarrow{v_i} = (v_{i2}, v_{i3}, \ldots, v_{iK}) = h(ID_i)$$

(13)

$$\sum_{j=2}^{K} v_{ij} = e$$

Entity registration processing:

When the centers 1 are requested by an entity i for registration, K secret vectors s_{ij} (j = 1, 2, ..., K)

corresponding, respectively, to a prepared key and K ID division vectors for entity i are found according to formulas 14-1, 14-2, ..., 14-K, as represented below, the vectors s_{ij} so found are sent to entity i in secret, and registration is complete.

(14-1) $$\overrightarrow{s_{i1}} \equiv g^{\alpha_i^{-e} H_1[\overrightarrow{I_{i1}}]} \pmod{P}$$

(14-2) $$\overrightarrow{s_{i2}} \equiv \alpha_i H_2[\overrightarrow{I_{i2}}]^{v_{i2}} \pmod{P-1}$$

$$\vdots$$

(14-K) $$\overrightarrow{s_{iK}} \equiv \alpha_i H_K[\overrightarrow{I_{iK}}]^{v_{iK}} \pmod{P-1}$$

Processing for generating common key between entities:

Entity i uses the disclosed hashing function h (·) to derive the second ID vector for an opposite entity m, namely v_m, according to formula 15 below.

(15) $$\overrightarrow{v_m} = (v_{m2}, v_{m3}, \ldots, v_{mK}) = h(ID_m)$$

Entity i selects from its own secret key vectors s_{i1} a vector s_{i1} [vector I_{m1}] of the component corresponding to vector I_{m1} that is the ID division vector of entity m, and also selects from among the secret key vectors s_{ij} for the blocks j (j = 2, ..., K) the vector s_{ij} [vector I_{mj}] of the component corresponding to the vector I_{mj}. Then, entity i sequentially performs power-multiplications, repeatedly for v_{mj} times, on all the vectors s_{ij} [vector I_{mj}] (j =

2, ..., K) except for vector s_{i1} [vector I_{m1}], with modulo P and the vector s_{i1} [vector I_{m1}] as the base, thereby deriving the common key K_{im}. The computation formula for finding this K_{im} specifically becomes formula 16 below. This K_{im} coincides with the common key K_{mi} obtained by the entity m.

(16)
$$\begin{aligned} K_{im} &\equiv \overrightarrow{s_{i1}}[\overrightarrow{I_{m1}}]^{\overrightarrow{s_{i2}}[\overrightarrow{I_{m2}}]^{v_{m2}} \cdots \overrightarrow{s_{iK}}[\overrightarrow{I_{mK}}]^{v_{mK}}} \\ &\equiv g^{\alpha_i^{-e}\alpha_i^{e} \cdot H_{1[i1][m1]} \cdot H_{2[i2][m2]}^{v_{m2}} \cdots H_{K[iK][mK]}^{v_{mK}}} \\ &\equiv g^{H_{1[i1][m1]} \cdot H_{2[i2][m2]}^{v_{i2}v_{m2}} \cdots H_{K[iK][mK]}^{v_{iK}v_{mK}}} \pmod{P} \end{aligned}$$

where $[\overrightarrow{I_{ij}}]$ is abbreviated $_{[ij]}$ from the second equation on

Safety against random number substitution attack:

Generally, in actual examples of the aforementioned entities A and B, we will have $v_{A2} \neq v_{B2}$, so that as shown below in formula 17, the random number substitution attack is not viable.

(17)
$$\begin{aligned} \overrightarrow{s_{Z2}}' &\equiv \frac{\overrightarrow{s_{A2}}[井]}{\overrightarrow{s_{B2}}[本]} \cdot \overrightarrow{s_{B2}} \\ &\equiv \frac{\alpha_A H_2[本][井]^{v_{A2}}}{\alpha_B H_2[井][本]^{v_{B2}}} \cdot \alpha_B H_2[井] \\ &\not\equiv \alpha_A H_2[井] \pmod{P-1} \end{aligned}$$

3rd Embodiment:

Another example (third embodiment) of the present invention is now described wherewith the personal random number elimination process is rendered complex by the addition of a constant term.

Preliminary processing at centers 1:

As in the first embodiment, the centers 1 prepare public keys and secret keys as follows and disclose the public keys.

Public key	N	N = PQ (where P and Q are large prime numbers)
	L	Size of ID vector (L = KM)
	K	Number of ID vector division blocks
	M	Size of divided ID vector
Secret key	g	Maximum generating element with modulo N
	H_j	Symmetrical $2^M \times 2^M$ matrix formed of random numbers ($j = 1, 2, \ldots, K$)
	α_{ij}	Personal secret random number of entity i where $\alpha_{i1}\alpha_i \ \alpha_{iK} \equiv 1 \pmod{\lambda(N)}$ and $\lambda(\cdot)$ is Carmichael function

Also, as in the first embodiment, the ID vector of each entity is divided into K blocks (ID division vectors) having a block size of M (cf. Fig. 2 and formula 1).

Entity registration processing:

When the centers 1 are requested by an entity i for registration, K secret vectors s_{ij} (j = 1, 2, ..., K-1, K) corresponding, respectively, to a prepared key and K ID division vectors for entity i are found according to formulas 18-1, 18-2, ..., 18-K-1, 18-K, as represented below.

(18-1) $$\overrightarrow{s_{i1}} \equiv g^{\alpha_{i1} H_1[\overrightarrow{I_{i1}}]} \pmod{N}$$

(18-2) $$\overrightarrow{s_{i2}} \equiv \alpha_i H_2[\overrightarrow{I_{i2}}] + \beta_{i2}$$

$$\vdots$$

(18-K-1) $$\overrightarrow{s_{i,K-1}} \equiv \alpha_i H_{K-1}[\overrightarrow{I_{i,K-1}}] + \beta_{i,K-1}$$

(18-K) $$\overrightarrow{s_{iK}} \equiv \alpha_{iK} H_K[\overrightarrow{I_{iK}}]$$

The third embodiment further adds K-2 personal random numbers β_{i2}, ..., $\beta_{i,K-1}$ to the first embodiment wherein $\alpha_{i2} = \ldots \alpha_{i,K-1} = \alpha_i$ and $\alpha_{i1}\alpha_i \alpha_{iK} = 1 \pmod{\lambda(N)}$. The centers 1 derive the vectors t_i according to formula 19 below. It should be assumed here that $\beta_i = \beta_{i2} + \ldots +$

$\beta_{i,K-1}$. The derived vectors s_{ij} and t_i are sent to entity i in secret and registration is complete.

$$(19) \qquad \overrightarrow{t_i} \equiv g^{-\alpha_{i1} H_1[\overrightarrow{I_{i1}}]\beta_i} \pmod{N}$$

Processing for generating common key between entities:

Entity i first, from the secret key vectors s_{ij} for the blocks j (j = 2, ..., K-1), selects column vectors s_{ij} [vectors I_{mj}] corresponding to the vectors I_{mj} that are the ID division vectors of entity m, block by block, and finds the sum S_{im} thereof by formula 20 below.

$$(20) \qquad \begin{aligned} S_{im} &= \sum_{j=2}^{K-1} \overrightarrow{s_{ij}}[\overrightarrow{I_{mj}}] \\ &= \alpha_i \sum_{j=2}^{K-1} H_j[\overrightarrow{I_{ij}}][\overrightarrow{I_{mj}}] + \beta_i \end{aligned}$$

Entity i, from among the secret key vector s_{i1} for its own first block and the secret key vector s_{iK} for the last block, selects the column corresponding to the vectors I_{mj} that are the ID division vectors of entity m, and performs the calculation shown below in formula 21 using s_{im} and vectors t_i to derive the common key K_{im}. This K_{im} coincides with the common key K_{mi} derived by entity m.

$$(21) \qquad \begin{aligned} K_{im} &\equiv \left(\overrightarrow{t_i}[\overrightarrow{I_{m1}}] \cdot \overrightarrow{s_{i1}}[\overrightarrow{I_{m1}}]^{S_{im}} \right)^{\overrightarrow{s_{iK}}[\overrightarrow{I_{mK}}]} \\ &\equiv g^{\alpha_{i1}\alpha_i\alpha_{iK} \cdot H_{1[i1][m1]} \left(\sum_{j=2}^{K-1} H_{j[ij][mj]} \right) H_{K[iK][mK]}} \end{aligned}$$

$$\equiv g^{H_{1[i1][m1]}\left(\sum_{j=2}^{K-1} H_{j[ij][mj]}\right) H_{K[iK][mK]}} \pmod{N}$$

where $\overrightarrow{[I_{ij}]}$ is abbreviated $_{[ij]}$ from the second equation on

Safety considerations:

In this formula, if settings are made as in formula 22 below, the expression $K_{im} = x_{im2}\ x_{im3}\ \ldots\ x_{im,K-1}$ will result, and, by gathering together numerous formulas wherein x_{im2}, x_{im3}, ..., $x_{im,K-1}$ are variables, it is theoretically possible to counterfeit keys.

(22)
$$\begin{aligned} x_{im2} &= g^{H_{1[i1][m1]}H_{2[i2][m2]}H_{K[iK][mK]}} \\ x_{im3} &= g^{H_{1[i1][m1]}H_{3[i3][m3]}H_{K[iK][mK]}} \\ &\vdots \\ x_{im,K-1} &= g^{H_{1[i1][m1]}H_{K-1[i,K-1][m,K-1]}H_{K[iK][mK]}} \end{aligned}$$

However, with the scheme of the present invention, the mathematical structures are held down to a minimum, and there is no structure in their variables that is separable, whereupon it becomes necessary to attack all of these variables as independent variables, thus requiring an extremely enormous number of collaborators. Even if the final block is susceptible to elimination by a random number substitution attack, the terms expressed

in formula 22 must be attacked as independent variables. Thus, in the case where M = 10, for example, it becomes necessary to amass 2^{20} specific equations in order to attack, so safety is enhanced.

Although the third embodiment pertains to a case wherein a composite number N difficult of prime factoring is used as the modulus, the same thing can of course be done in the case where N = P.

4th Embodiment:

Fig. 4 is a model diagram showing the configuration of a cryptographic communications system of the present invention. A plurality (K) of centers 1 which can be trusted to maintain information confidentiality are established. These centers 1 may be public institutions in a society, for example.

These centers 1 and a plurality of entities a, b, ..., z that are users of this cryptosystem are connected by secret communication paths 2_{a1}, ..., 2_{aK}, 2_{b1}, ..., 2_{bK}, ..., 2_{z1}, ..., 2_{zK}. Thus secret key information can be sent to the entities a, b, ..., z from the centers 1 via the secret communication paths. Communication paths 3ab, 3az, 3bz, etc., are also deployed between pairs of entities so that ciphertext resulting from encrypting communications information can be sent back and forth between entities via those communication paths 3ab, 3az, 3bz, etc.

Preparatory processing at centers 1:

The centers 1 prepare public keys and secret keys as shown below, and discloses the public keys.

Public key	N	$N = PQ$
	K	Number of ID vector division blocks
	M_j	Size of divided ID vector (where $j = 1, 2, \ldots, K$)
	L	Size of ID vector ($L = M_1 + M_2 + \ldots + M_K$)
	T	Degree of exponent portion
Secret key	P,Q	Large prime numbers
	g	Maximum generating element with modulo N
	H_j	Symmetrical $2^{Mj} \times 2^{Mj}$ matrix formed of random numbers
	α_i	Personal secret random number of entity i (where gcd (α_i, $\lambda(N)$) = 1 and $\lambda(\cdot)$ is Carmichael function)
	β_{ij}	Personal secret random number of entity i (where $\beta_{i1} + \beta_{i2} + \ldots + \beta_{iK} = \lambda(N)$)

It should be assumed that ID vectors that are specifying information indicating the names and addresses

of entities are L-dimension binary vectors, and each of their ID vectors is divided into K blocks (block sizes are M_1, M_2, ..., M_K), as diagrammed in Fig. 5. The ID vector for entity i (i.e. vector I_i), for example, is divided as indicated in formula 23 below. The vectors I_{ij} (j = 1, 2, ..., K), that are divided specifying information, are called ID division vectors.

$$(23) \qquad \overrightarrow{I_i} = [\overrightarrow{I_{i1}}|\overrightarrow{I_{i2}}|\cdots|\overrightarrow{I_{iK}}]$$

Entity registration processing:

When the centers 1 are requested by an entity i for registration, K secret key vectors s_{ij} (j = 1, 2, ..., K) corresponding, respectively, to a prepared key and K ID division vectors for entity i are calculated according to formulas 24-1, 24-2, ..., 24-j, ..., 24-K below.

$$(24\text{-}1) \qquad \overrightarrow{s_{i1}} = \alpha_i H_1[\overrightarrow{I_{i1}}] + \beta_{i1}\overrightarrow{1}$$

$$(24\text{-}2) \qquad \overrightarrow{s_{i2}} = \alpha_i H_2[\overrightarrow{I_{i2}}] + \beta_{i2}\overrightarrow{1}$$

$$\vdots$$

$$(24\text{-}j) \qquad \overrightarrow{s_{ij}} = \alpha_i H_j[\overrightarrow{I_{ij}}] + \beta_{ij}\overrightarrow{1}$$

$$\vdots$$

$$(24\text{-}K) \qquad \overrightarrow{s_{iK}} = \alpha_i H_K[\overrightarrow{I_{iK}}] + \beta_{iK}\overrightarrow{1}$$

Vector 1 represents a vector of K dimension wherein all of the components are 1. The representation H_j [vector I_{ij}] indicates a row, corresponding to the vector

I_{ij}, extracted from the symmetrical matrix H_j, and the $[\cdot]$ operation is also defined for reference.

Next, for the 1st block, T + 1 secret key vectors g_{it} (t = 0, 1, 2, ..., T) are calculated according to formulas 25-0, 25-1, 25-2, ..., 25-t, ..., 25-T below.

$$(25\text{-}0) \qquad \overrightarrow{g_{i0}} \equiv g^{\alpha_i^{-T}\overrightarrow{1}} \pmod{N}$$

$$(25\text{-}1) \qquad \overrightarrow{g_{i1}} \equiv g^{\alpha_i^{-T}\overrightarrow{s_{i1}}} \pmod{N}$$

$$(25\text{-}2) \qquad \overrightarrow{g_{i2}} \equiv g^{\alpha_i^{-T}\langle\overrightarrow{s_{i1}}\rangle^2} \pmod{N}$$

$$\vdots$$

$$(25\text{-}t) \qquad \overrightarrow{g_{it}} \equiv g^{\alpha_i^{-T}\langle\overrightarrow{s_{i1}}\rangle^t} \pmod{N}$$

$$\vdots$$

$$(25\text{-}T) \qquad \overrightarrow{g_{iT}} \equiv g^{\alpha_i^{-T}\langle\overrightarrow{s_{i1}}\rangle^T} \pmod{N}$$

It should be assumed that when c is a scalar and A and B indicated in formulas 26 and 27 are matrixes, the expressions $B = c^A$ and $B = \langle A\rangle^c$ correspond to formulas 28 and 29, respectively.

$$(26) \qquad A = (a_{\mu\nu})$$

$$(27) \qquad B = (b_{\mu\nu})$$

$$(28) \qquad b_{\mu\nu} = c^{a_{\mu\nu}}$$

$$(29) \qquad b_{\mu\nu} = a_{\mu\nu}{}^c$$

One of the centers 1 sends the T + 1 secret key vectors g_{it} (t = 0, 1, 2, ..., T) relating to 1st block to entities i in secret, while the remaining (K - 1) centers 1 send K - 1 secret key vectors s_{ij} (j = 2, 3, ..., K) relating to the blocks from the second to the last to entities i in secret.

Processing for generating common key between entities:

Entity i, for the 1st block, selects from its own T + 1 secret key vectors g_{it} a vector g_{it} [vector I_{m1}] of the component corresponding to vector I_{m1} that is the ID division vector of entity m. The vectors selected are represented below in formulas 30-0, 30-1, ..., 30-t, ..., 30-T.

(30-0) $$g_{0im} = \overrightarrow{g_{i0}}[\overrightarrow{I_{m1}}]$$

(30-1) $$g_{1im} = \overrightarrow{g_{i1}}[\overrightarrow{I_{m1}}]$$

$$\vdots$$

(30-t) $$g_{tim} = \overrightarrow{g_{it}}[\overrightarrow{I_{m1}}]$$

$$\vdots$$

(30-T) $$g_{Tim} = \overrightarrow{g_{iT}}[\overrightarrow{I_{m1}}]$$

Next, entity i, for the blocks 2, 3, ..., K for j = 2, 3, ..., K, selects, from its own secret key vectors s_{ij}, vectors s_{ij} [vectors I_{mj}] of the components corresponding to vectors I_{mj} that are the ID division vectors of entity

m, block by block. The vectors selected are represented below in formulas 31-2, ..., 31-J, ..., 31-K.

$$(31\text{-}2) \qquad x_{2im} = \overrightarrow{s_{i2}}[\overrightarrow{I_{m2}}]$$

$$\vdots$$

$$(31\text{-}j) \qquad x_{jim} = \overrightarrow{s_{ij}}[\overrightarrow{I_{mj}}]$$

$$\vdots$$

$$(31\text{-}K) \qquad x_{Kim} = \overrightarrow{s_{iK}}[\overrightarrow{I_{mK}}]$$

Then, the sum y_{im} for all of these is found on the integer ring as in formula 32 below.

$$(32) \qquad y_{im} = \sum_{j=2}^{K} x_{jim}$$

And, by performing calculation as in formula 33 below, with modulo N, the common key K_{im} is derived. In the calculation in this formula 33, by completing the calculations for all of the blocks, the personal secret random number α_i is eliminated by multiplication by the inverse element thereof, and the personal secret random numbers β_{ij}, which are K in number, are eliminated by additions therefor. This K_{im} coincides with the common key K_{mi} derived by entity m.

$$(33) \qquad \begin{aligned} K_{im} &\equiv \prod_{t=0}^{T} g_{tim}^{{}_{T}C_t y_{im}^{(T-t)}} \\ &\equiv g^{\alpha_i^{-T} \sum_{t=0}^{T} {}_{T}C_t x_{1im}^{t} y_{im}^{T-t}} \\ &\equiv g^{\alpha_i^{-T}(x_{1im}+y_{im})^T} \\ &\equiv g^{\alpha_i^{-T}(x_{1im}+\cdots+x_{Kim})^T} \end{aligned}$$

$$\equiv g^{\alpha_i^{-T}}(\alpha_i H_1[\overrightarrow{I_{i1}}][\overrightarrow{I_{m1}}] + \beta_{i1} + \cdots + \alpha_i H_K[\overrightarrow{I_{iK}}][\overrightarrow{I_{mK}}] + \beta_{iK})^T$$

$$\equiv g^{\alpha_i^{-T}}\left\{\alpha_i(H_1[\overrightarrow{I_{i1}}][\overrightarrow{I_{m1}}] + \cdots + H_K[\overrightarrow{I_{iK}}][\overrightarrow{I_{mK}}]) + \lambda(N)\right\}^T$$

$$\equiv g^{\alpha_i^{-T}}\left\{\alpha_i(H_1[\overrightarrow{I_{i1}}][\overrightarrow{I_{m1}}] + \cdots + H_K[\overrightarrow{I_{iK}}][\overrightarrow{I_{mK}}])\right\}^T$$

$$\equiv g(H_1[\overrightarrow{I_{i1}}][\overrightarrow{I_{m1}}] + \cdots + H_K[\overrightarrow{I_{iK}}][\overrightarrow{I_{mK}}])^T \pmod N$$

In the formula above we assumed x_{1im} = vector s_{i1} [vector I_{m1}], but this is not even known to entity i itself. Also, because T is a comparatively small number, the exponent portion can be calculated by successively and repeatedly performing power multiplication.

In the example described in the foregoing, the size M_j of the blocks may be constant for all blocks or, alternatively, some or all of the blocks may have different sizes. However, the secret key vector g_{it} is derived in relation to the 1st block, wherefore, when that size is made constant for all blocks, the secret becomes large for the 1st block. Thus, it is better to make the size of the 1st block smaller than the sizes of the other blocks. When $M_1 = 1$, in particular, the secrets distributed can be minimized and safety most enhanced.

Let us now consider the safety of the present invention against a collusive attack such as an attack against the whole cryptosystem by the collusion of a large indefinite number of entities. If the total number of entities is 1 million, then $1000000 \fallingdotseq 2^{20}$, wherefore M_j

= 1 and K = 20. If T = 32, then the number of exponent portion terms in the common key K_{im} becomes ${}_{20}H_{32} = {}_{51}C_{32} \fallingdotseq 4.85 \times 10^{13}$. This number of terms exceeds the total number of keys shared between all entities, namely ${}_{1000000}C_2 \fallingdotseq 5 \times 10^{12}$. Accordingly the condition that number of terms > total number of shared keys is satisfied and safety against collusive attack is realized.

The communication of information between entities in the cryptosystem described in the foregoing is described next. Fig. 6 is a model diagram showing how information is communicated between two entities a and b. In the example diagrammed in Fig. 6, entity a encrypts a plaintext (message) M to the ciphertext C which it sends to entity b, and entity b decrypts that ciphertext C back to the original plaintext (message) M.

The first of the centers 1 is equipped with a secret key generator 1a which computes secret key vectors s_{a1} and s_{b1} peculiar to the entities a and b, and the secret key vectors g_{at} and g_{bt} (t = 0, 1, 2, ..., T) numbering T + 1, according to the formulas 24-1, 25-0, ..., 25-T given earlier. Then, when registration requests are made by the entities a and b, the secret key vectors g_{at} and g_{bt} for those entities a and b are sent to the entities a and b.

The j'th center 1 (where j = 2, 3, ..., K) is equipped with a secret key generator 1a for computing the secret key vectors s_{aj} and s_{bj} peculiar to the entities a and b according to the formulas 24-2, ..., 24-K given earlier. When registration requests are made by the entities a and b, the secret key vectors s_{aj} and s_{bj} for those entities a and b are sent to the entities a and b.

Entity a is provided with a memory 10 for storing, in tabular form, the secret key vectors g_{at} (t = 0, 1, 2, ..., T) and s_{aj} (j = 2, 3, ..., K) sent from the centers 1, a component selector 11 for selecting from among those secret key vectors the vector g_{at} [vector I_{b1}] (t = 0, 1, 2, ..., T) and the vector s_{aj} [vector I_{bj}] (j = 2, 3, ..., K) for the components corresponding to entity b, a common key generator 12 for generating the common key K_{ab} with entity b derived by entity a using those components selected, and an encryptor 13 for encrypting the plaintext (message) M to the ciphertext C using the common key K_{ab} and outputting it over the channel 30.

Entity b is provided with a memory 20 for storing, in tabular form, the secret key vectors g_{bt} (t = 0, 1, 2, ..., T) and s_{bj} (j = 2, 3, ..., K) sent from the centers 1, a component selector 21 for selecting from among the secret key vectors the vector g_{bt} [vector I_{a1}] (t = 0, 1, 2, ..., T) and the vector s_{bj} [vector I_{aj}] (j = 2, 3, ..., K) for the components corresponding to entity a,

a common key generator 22 for generating the common key K_{ba} with entity a derived by entity b using those components selected, and a decryptor 23 for decrypting the ciphertext C input from the channel 30 to the plaintext M using the common key K_{ba} and outputting it.

When information is to be sent from entity a to entity b, first, the secret key vectors g_{at} (t = 0, 1, 2, ..., T) and s_{aj} (j = 2, 3, ..., K) pre-stored in the memory 10 after being derived at the centers 1 are read out to the component selector 11. The component selector 11 then selects the vector g_{at} [vector I_{b1}] (t = 0, 1, 2, ..., T) and the vector s_{aj} [vector I_{bj}] (j = 2, 3, ..., K) that are the components corresponding to entity b and sends them to the common key generator 12. The common key generator 12 uses these components to derive the common key K_{ab} according to formula 33, and sends the common key K_{ab} to the encryptor 13. The encryptor 13 utilizes this common key K_{ab} to encrypt the plaintext M to the ciphertext C and sends the ciphertext C via the channel 30.

The ciphertext C sent over the channel 30 is input to the decryptor 23 of entity b. The secret key vectors s_{bj} (j = 2, 3, ..., K) and g_{bt} (t = 0, 1, 2, ..., T) derived at the centers 1 and prestored in the memory 20 are read out to the component selector 21. Then, the component selector 21 selects the vector g_{bt} [vector I_{a1}]

(t = 0, 1, 2, ..., T) and the vector s_{bj} [vector I_{aj}] (j = 2, 3, ..., K) that are components corresponding to entity a and sends them to the common key generator 22. The common key generator 22 uses these components to derive the common key K_{ba} according to formula 33 and sends this common key to the decryptor 23. The decryptor 23 uses the common key K_{ba} to decrypt the ciphertext C to the plaintext M.

In the above-described example, centers are deployed in a plurality, and these centers generate a plurality of keys corresponding to a plurality of units (pieces) of entity ID information respectively. In other words, each center generates a key for a certain segment of entity ID information. Therefore no single center can hold all entity secrets, and the centers cannot become "big brothers." Also, the secret key vectors peculiar to the respective entities are stored beforehand in the memories of the entities, so the time required for generating common keys can be shortened.

Fig. 7 is a configurational diagram of an embodiment of recording media according to the present invention. The program exemplified here, which is recorded on recording media described below, comprises processes for selecting components corresponding to entity m from among the secret key vectors s_{ij} and g_{it} sent to entity i from the centers and processes for finding a common key K_{im}

using those components so selected. A computer 40 is provided at each entity.

In Fig. 7, a recording medium 41 that connects the computer 40 online employs a WWW (world wide web) server computer, for example, located remotely from the site where the computer 40 is installed. A program 41a such as that described above is recorded on the recording medium 41. The program 41a read out from the recording medium 41 controls the computer 40 and thereby computes common keys at the entities for other entities to be communicated with.

A recording medium 42 provided internally in the computer 40 is a built-in hard disk drive or ROM, for example, and a program 42a as described above is recorded on the recording medium 42. The program 42a read out from the recording medium 42 controls the computer 40 and thereby computes common keys at the entities for other entities to be communicated with.

A recording medium 43 loaded in a disk drive 40a of the computer 40 is a portable optical-magnetic disk, CD-ROM, or flexible disk, etc. A program 43a such as described above is recorded on the recording medium 43. The program 43a retrieved from the recording medium 43 controls the computer 40 and thereby computes common keys at the entities for other entities to be communicated with.

With the present invention, as described in the foregoing, entity ID information is divided into a plurality of segments or pieces and a plurality of centers are established for these entity ID information pieces respectively such that each of the centers generates a particular key for a particular piece of entity ID information. Therefore, no single center can grasp all entity secrets or can become a "big brother." In addition, the mathematical structures are held down to a minimum, so that it is easy both to effectively circumvent the collusion problem and to implement the cryptosystem. Furthermore, because the entities are in possession beforehand of secret keys peculiar thereto, the time required for generating common keys can be significantly shortened.

With an ID-NIKS based on the third conventional method described earlier, in general, L × L symmetrical matrixes are center secrets, and a portion of that information is treated as a vector comprising L components and distributed to the entities. This scheme is very easy to implement but the collusion threshold is no more than approximately L. With the scheme of the present invention, on the other hand, a collusion threshold can be obtained which is far greater than L.

With the conventional scheme, by employing $2^M \times 2^M$ center secret matrixes, it is possible to configure an ID-NIKS having the same level of collusion threshold as the present invention. An ID-NIKS configured in such manner is not practical, however, because it requires 2^M product computations or power-multiplication computations for key sharing. Another problem with such an ID-NIKS is that almost all schemes are divisible so that secret keys can be counterfeited for entities expressed by the linear combination of some collaborators. With the scheme of the present invention, on the other hand, the number of secret keys held becomes more numerous, but the common keys can be shared by making K-1 power-multiplication computations, at most, key generation can be done at very high speed, and, even though some entities might be expressed by the linear combination of collaborators, it is still possible to prevent the counterfeiting of secret keys for those entities.

With the present invention, moreover, the random number terms are eliminated only after all blocks have been completely computed, wherefore divided blocks cannot be independently attacked and it is possible to circumvent random number substitution attack.

The above illustrated and described secret key generation method, encryption method, cryptographic communications method, common key generator,

cryptographic communications system, and recording media are disclosed in Japanese Patent Application Nos. 11-16257 and 11-59049 filed on January 25, 1999 and March 5, 1999 respectively, the instant application claims priority of these Japanese Applications, and the entire disclosure thereof is herein incorporated by reference.